REMARKS

Remaining Claims

Twenty (20) claims (Claims 1-20) remain pending in this application through this Amendment. Claims 1-9 and 16-20 have been rejected by the Examiner. Claims 10–15 have been objected to by the Examiner as depending from rejected claims but otherwise allowable. The Applicant has amended claim 20 to correct an informality noted by the Examiner. The Applicant respectfully traverses the remaining rejections and requests further examination and reconsideration of the application.

Allowable Subject Matter

The Applicant wishes to thank the Examiner for recognizing the allowable subject matter of claims 10-15.

Rejection of Claim 20 under 35 U.S.C. §112, ¶ 2

Claim 20 stands rejected under 35 U.S.C. §112, second paragraph. The Applicant has amended claim 20 to remove the word "not" in the second condition recited in the claim so that the claim properly recites two different conditions and their respective results. The Applicant believes the correction of this informality overcomes the rejection.

Rejection of Claims 1-9 and 16-20 under 35 U.S.C. §102(e) – DeLano

Claims 1-9 and 16-20 stand rejected under 35 U.S.C. §102(e) as being anticipated by *DeLano* (U.S. Patent Application Publication No. 2003/0163763). The Applicant respectfully traverses this rejection for at least the following reasons.

DeLano plainly does not disclose any type of "FIFO," which is central to every one of the Applicant's claims. The term "FIFO" stands for "first-in first-out" and is well-understood in the art to refer to a queue-like storage structure in which data values are read out in the same order in which they were stored; that is, the first value stored in the FIFO will be the first value read out of the FIFO. At least conceptually, a FIFO has one data input, one data output, and a queue of memory locations between the input and output. As each successive value is received at the input and stored, the previously-stored values make their way through the queue toward the output. A FIFO can be thought of as analogous to a bookshelf, in which books are slid onto the shelf at one end (the input), and the books previously put on the shelf are pushed toward the other end (the output) by each successive book that is slid onto the input end. The first book that was put on the shelf at the input end will be the first book that can be taken off the shelf at the output end.

It is such a concept that is central to the Applicant's invention because, in essence, the Applicant has invented something analogous to a way of putting a book back on the shelf (at the <u>output</u> end from which it was taken), in the event the book was inadvertently taken off too soon. In a conventional FIFO, a data value can <u>only</u> be stored by placing it at the input, and it must make its way through the queue to the output before it can be read out again.

The Applicant respectfully submits that what the Examiner characterizes in Fig. 1 of *DeLano* as a FIFO (element 12) is not a FIFO at all. Element 12, which *DeLano* refers to as a "register file," has multiple inputs served by a "write multiplexer (mux)" and multiple outputs served by a "read multiplexer (mux)." As stated in paragraph 16 of *DeLano*, "Register file 12 has multiple write ports processed through a write mux 14, and multiple read ports processed through a read mux 16." A data value received at the input can be directed by the write mux into <u>any</u> selected one of the 128 storage registers. Likewise, a data value can be read out from <u>any</u> selected one of the 128 storage registers as directed by the read mux.

In contrast, a conventional FIFO can be thought of as having only a single write port or input and a single read port or output. In a <u>FIFO</u>, at any given time, only the earliest data value that was written in to the FIFO may be read out of the FIFO; data values that have been written in to the FIFO at other times cannot be read out. Stated another way, data values cannot be written to or read from a storage register in the middle of data stored in the FIFO because there are no write and read ports for doing so. The essential algorithm or sequence that characterizes the operation of a FIFO is first-in first-out. FIFO memory is rigidly controlled (by the first-in first-out rule), in stark contrast to the freely or randomly accessible collection of storage registers of a <u>register file</u> such as that shown in *DeLano*.

Nowhere in *DeLano* is a first-in first-out sequence described for writing and reading data values to and from the 128 registers. *DeLano* states no more than: "In summary, prior data of a particular register is stored within buffer 20 prior to a register load of that register within register file 20, via read port 18 and bus 60, just prior to architecting the new data within the register of register file 12, e.g., at a write-back stage through bus 32." Such a statement does not in any way imply that data values are written to or read from the registers of register file 12 in any particular sequence, much less the very particular sequence that characterizes a FIFO of data values being read out in the same order in which they were written in. Indeed, the very presence of a write mux and a read mux implies the capability of freely or randomly writing to and reading from registers in the middle of the register file. A data value can be written to any selected one of the 128 registers by selecting it via the write mux. Likewise, a data value can be read from any selected one of the 128 registers by selecting it via the read mux. In summary, the register file of *DeLano* is a bank or file of randomly accessible registers, not a queue that operates under a first-in first-out rule.

Although *DeLano* relates to recovering from a corrupted speculative write to the register file by storing a copy of the data in a buffer (20) and rewriting the register file with the data copy if there was an error, the register file is not a FIFO. Rather, the data copy can be rewritten to any selected register in the register file.

The Applicant's invention relates specifically to a novel FIFO in which a data value, if read out prematurely, can be, in effect, "pushed back in." In a conventional FIFO, a value cannot be pushed back in because any data value that is written to the FIFO can only be read out after all previously-written data values have been read out. Only in the Applicant's "pushback FIFO" can a data value be, in effect, pushed back in, such that it can be immediately read out again. *DeLano* does not disclose a FIFO of any type, much less anything like the Applicant's pushback FIFO. Therefore, *DeLano* does not anticipate any of the Applicant's claims, and the Applicant respectfully requests reconsideration and withdrawal of the rejection under 35 U.S.C. §102(e) of claims 1-9 and 16-20.

CONCLUSION

For the reasons set forth above, it is respectfully submitted that all pending claims are now in condition for allowance, and the Applicant requests a Notice of Allowance be issued in this case. Should there be any further questions or concerns, the Examiner is urged to telephone the undersigned to expedite prosecution.

Respectfully submitted,
GARDNER GROFF, P.C.

Lawrence D. Maxwell
Reg. No. 35,276

GARDNER GROFF, P.C.
Paper Mill Village, Building 23
600 Village Trace, Suite 300
Marietta, Georgia 30067
Phone: 770.984.2300
Fax: 770.984.0098